Project Manager Introduction

Welcome to CodeWarrior.

This tutorial is designed to introduce the basic features and functionality of the Project Manager.

The Project Manager is a major part of the CodeWarrior Integrated Development Environment (IDE).

Using the Controls

Use these three controls to move through a tutorial.

Previous

Click to move to the previous page.

Return to Topics

Click to return to the main topics page.

Next

Click to move to the next page.

Adobe Acrobat

The standard Adobe Acrobat keyboard commands are also available.

Project Manager Overview

The topics covered in this introduction include:

- Using the Controls
- Project Manager Topics
- Required Knowledge
- Intended Audience
- Taking the Tour
- Host Conventions
- Where Else to Go Tutorials
- Where Else to Go Documentation

Project Manager Topics

This tutorial has two purposes. First, it introduces the CodeWarrior Project Manager features. Second, it guides you through several common actions you need to know to use the CodeWarrior IDE. This tutorial will cover:

- Project Manager Components
- STAGE 1: Touring the File View
- STAGE 2: Touring the Link Order View
- STAGE 3: Touring the Targets View
- STAGE 4: Touring Toolbars
- Project Manager Actions
- STAGE 5: Using File Actions
- STAGE 6: Using Project Actions

Required Knowledge

This tour assumes that you have a basic understanding of using your computer system. If you are just getting started with computers, refer to the user guide that came with your computer system. Once you feel comfortable with using your computer, come back and start again.

NOTE: The CodeWarrior tutorials are designed to teach you how to use CodeWarrior to successfully create, compile, and debug executable code. These tutorials will not teach you how to program a computer. If you wish to learn programming, visit your local bookstore for beginning programming books, see your local community college for classes on programming, or refer to the 3rd-party documentation that may be provided on your CodeWarrior CD.

Intended Audience

The Project Manager (Basic) tutorial is intended for all first-time code warriors. If you have not already taken the Guided Tour tutorial, we recommend that you take a few minutes to go through that tutorial before continuing here. The Guided Tour will give you an overview of the entire CodeWarrior IDE.

Beginning Programmers

If you are just beginning your career as a programmer then you are in for a treat. The CodeWarrior environment provides all the tools you need for any programming task, from simple applications to advanced professional-quality programs. Go through this tour, then follow the recommendations at "Where Else to Go Tutorials" to continue your explorations of CodeWarrior.

more…

Intended Audience (continued)

Experienced Programmers

After taking the Guided Tour tutorial, experienced programmers new to CodeWarrior should refer to this and the other tutorials for in-depth knowledge about the various components of the CodeWarrior environment. Many CodeWarrior Targeting manusl include basic tutorials that demonstrate how to use the tools to program for a particular platform.

Taking the Tour

The CodeWarrior tutorials are designed to assist you in becoming familiar with the CodeWarrior IDE. This tutorial consists of a series of stages. Each stage, in turn, is a series of slides, like this one. Each slide describes a feature or a task you can perform. You will gain the most benefit from these tutorials if you follow along with the CodeWarrior IDE while reading and performing the actions described in the tutorial.

To Use a Tutorial:

1. In Acrobat Reader 3.0, read the text on a slide.
2. If actions are described (example: Choose Open from the File menu), then switch to the CodeWarrior IDE.
3. In the CodeWarrior IDE, perform the action.
4. Switch back to Adobe Acrobat Reader, read the next action.

NOTE: Both Adobe Acrobat Reader and the CodeWarrior IDE must be running.

Host Conventions

Hosts: While the CodeWarrior IDE runs on several different host platforms, the functionality on these host platforms is not always identical. There are occasions when a feature is implemented for one or more hosts and not for others. This tutorial only demonstrates features found on all supported hosts.

Pictures: The IDE's visual interface on these hosts is nearly identical in all significant respects. When discussing a particular interface element such as a dialog box or window, the screenshot may come from any host. If you are using CodeWarrior on a different host, you should have no difficulty understanding the picture.

Keyboard: The default keyboard shortcuts for CodeWarrior on these platforms are very similar. However, keyboards and shortcuts do vary across host platforms. For example, a typical keyboard for a Windows machine has an Alt key, but a typical keyboard for a Mac OS computer calls the same key the Option key

more…

Host Conventions (continued)

To handle these kinds of situations, CodeWarrior documentation identifies and uses the following paired terms in the text:

- `Enter/Return` - the “carriage return” or “end of line” key. This is not the numeric keypad Enter key, although in almost all cases that works the same.
- `Backspace/Delete` - the key that (in text editing) causes the character before the insertion point to be erased (not to be confused with `Delete/Del`, the “forward delete” key)
- `Ctrl/Command` - in most cases, CodeWarrior maps the Windows Ctrl key the same way as the Mac OS Command key.
- `Alt/Option` - in most cases, CodeWarrior maps the Windows Alt key the same way as the Mac OS Option key.

For example, you may encounter instructions such as “Press `Ctrl/Command+S` to save a file,” or “Alt/Option click the Function pop-up to see the functions in alphabetical order." Use the appropriate key as it is labeled on your keyboard.

Where Else to Go Tutorials

Once you have finished this tutorial, be sure to explore the other CodeWarrior tutorials, and the documentation that accompanies this product. For an overview of the IDE, see:

- Guided Tour–overview of the CodeWarrior IDE
- Project Manager (Basic)–introduces the Project Manager
- Editor (Basic)–introduces the CodeWarrior Editor
- MW Debugger (Basic)–introduces debugging features
- Find & Replace–introduces Find and Replace features

Where Else to Go Documentation

To learn more about programming for a particular target operating system or processor, see the corresponding CodeWarrior Targeting manual. The CodeWarrior Project Manager features are also documented in:

- IDE User Guide–documents IDE components, including the Project Manager.
- Targeting manual for your programming target. Examples include Targeting Windows, Targeting Rhapsody, and Targeting Palm OS.

Getting Started

Whew, now that all of the preliminaries are out of the way, it's time to explore the Project Manager.

Project Manager Components

This section of the tutorial identifies the components of the Project Manager window and describes their use.

Along the way we will cover:

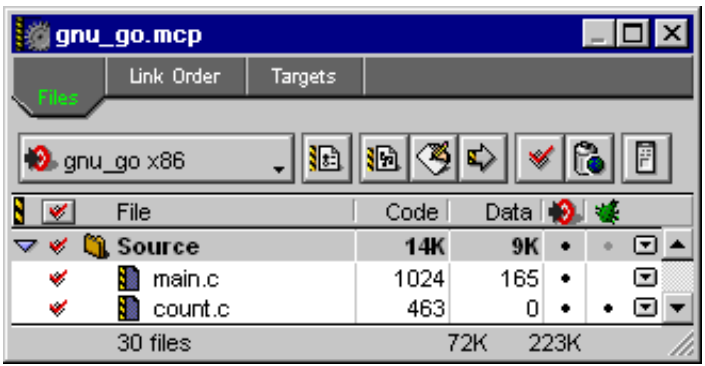
- STAGE 1: Touring the File View
- STAGE 2: Touring the Link Order View
- STAGE 3: Touring the Targets View
- STAGE 4: Touring Toolbars

Getting Ready

You will use the `gnu_go.mcp` project file throughout this tutorial. If you don't already have it open, please open the project now.

The `gnu_go.mcp` project file is located in the `Source Examples` folder inside the `Core Tutorials` folder.

If you don't know how to open a project file, see the Guided Tour tutorial.



Touring the File View

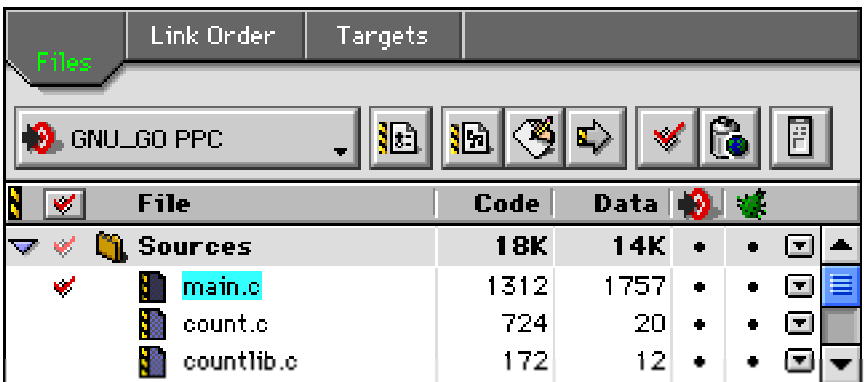
The first view to look at in a project window is the Files view. This is the default view that appears when you create a project.

The topics include:

- Files View Introduction
- File View Columns
- File View Groups
- File View Pop-ups

Files View Introduction

The Files view enables you to inspect file information in a project. This information includes the name of source files, groups, and status information on each.



File View Columns

The File view contains several columns for displaying information about project files. These columns (from left to right in the Project window) include:

Touch–identifies files that require compilation the next time the project is built.

File–lists the name of each file and group.

Code–shows the size of the compiled executable code for each file.

Data–shows the size of the non-executable data area for each file.

Targets–indicates whether a file belongs to the current target.

Debug–indicates whether debugging information is being generated for the file.

VCS–indicates the current checkout status of the file. This column is only present when a version control system (VCS) is activated.

File View Groups

Groups provide a way to organize project files into logical categories to help you keep track of your files. Groups are identified by the gray bar and disclosure triangle that appear with the group name.

Group topics include:

- Creating Groups
- Moving Groups
- Renaming Groups
- Deleting Groups

NOTE: You can click the disclosure triangle to the left of a group name to expand (Ctrl/Command+Right Arrow) or collapse (Ctrl/Command+Left Arrow) the group.

Creating Groups

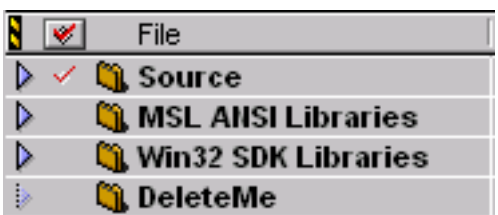
It's easy to create groups in a project. This lets you keep your files organized as you add files to the project.

To Create a Group:

1. Choose **Create New Group** from the **Project** menu.

   The Create Group dialog box appears.
2. Enter `DeleteMe` in the **Create Group** dialog.
3. Click **OK**.

   The `DeleteMe` group appears in the Project window.



Moving Groups

As projects expand and the number of groups increases, you can organize the groups for a better understanding of their project relationship. Groups can be moved anywhere on the same level, or into another group by noting the position of the small arrowhead on the focus line.

To Move a Group:

- Drag the group to a new position, using the focus line to indicate the drag destination, and release the mouse button to end the drag.

  Do this now with the `DeleteMe` group created earlier. Drag it to a new location and observe what happens.

NOTE: This also works with files.

Renaming Groups

You may decide to rename a group to better identify the files that reside within.

To Rename a Group:

1. Double-click the group in the project window, **OR**
   Select the group, then press the Enter key.

   The Rename Group dialog box appears.
2. Enter the new name in the **Rename Group** dialog.
3. Click **OK**.

   The new group name appears in the Project window.

Deleting Groups

Entire groups can sometimes become obsolete. When that occurs you can delete a group from a project quickly and easily.

To Delete a Group:

1. Select the group to delete.
2. Choose **Remove Selected Items** from the **Project** menu, **OR**
   Type the keyboard shortcut Ctrl/Option+Delete.

   The selected group *and all its files* are removed from the Project window.

NOTE: This action is not undoable.

File View Pop-ups

Pop-up menus in the File view reveal additional information about the group or file associated with it. The two types of interface pop-ups include:

Group Pop-ups–display a list of all files contained in that group.

Interfaces Pop-up–displays a list of interface (header) files required to compile the file, and allows you to 'touch' the file, forcing it to recompile on the next build.

To View a Pop-up:

- Click on the pop-up menu to the right of the file or group name.

  The interface group pop-up appears. You can open a file that appears in either pop-up simply by choosing it from the menu.

Context Pop-ups

The two forms of Context Pop-ups available in the Project window include:

Group Context Pop-up–enables you to Check Syntax, Preprocess, Precompile, Compile, Disassemble, or Remove Selected Items.

Context Pop-up–does everything the Group Context Pop-up does as well as showing the File Path, and Reveal in Finder the file's location.

To View a Context Pop-up:

1. Position the cursor over a file or group name.
2. Click and hold the mouse button down.

   The Context pop-up appears.

TIP: On Windows you can also use the right mouse button.

Summary

This concludes the File View stage of this tutorial.

In this stage you learned how to:

- Identify the File view
- Identify the columns within the File view
- Create, rename, move, and delete groups
- Use the Interface and context pop-ups within the Project window

In the next stage you explore another view in the Project window, the Link Order view. It has many similarities with the File view.

Touring the Link Order View

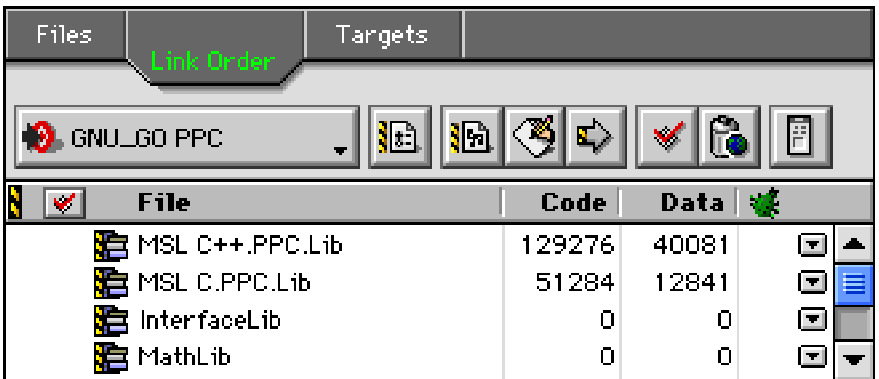
The second view in a Project Manager window is the Link Order view.

The slides for the Link Order view include:

- Link Order View Introduction
- Setting the Link Order
- The Segments View

Link Order View Introduction

The Link Order view provides file compiling and linking order information when using C++. Pascal and Java use their own linking order. The Link Order view only displays files built and used by the current target.



Setting the Link Order

Use the Link Order view to specify the compile and linking order of your C++ source files. Pascal and Java use their own linking order. This is especially handy if your project is dependent upon static initializers being setup before the code from other files run.

To Set the Link Order:

- Drag the file to a new position, using the focus line to indicate the drag destination, and release the mouse button to end the drag.

  Do this now with the `main.c` file. Drag it to a new location and observe what happens.

The Segments View

The Segments view replaces the Link Order view when a MacOS 68K project is present.

The Segments view provides segmentation information on files used when compiling and linking MacOS 68K projects. A single segment in a 68K project cannot exceed 32K in size. The Segments view allows you to specify which files are compiled into which segments.

For More Information

See the Targeting MacOS manual for details on using Segments.

Summary

This concludes the Link Order stage of this tour.

In this stage you learned how to:

- Identify the Link Order view
- Set the link order of files
- Identify the Segments view

In the next stage you explore the final view in the Project window, the Target view. This view serves a very different purpose–target management rather than file management.

Touring the Targets View

The third view in the Project Manager window is the Targets view.

The slides for the Targets view include:

- Targets View Introduction
- Targets View Columns
- Creating Targets
- Renaming Targets
- Deleting Targets

Targets View Introduction

The Target view displays information about the different targets in a project. For example, you can have a debug build and a release build. A target is a separate build you can generate from the files contained in the project. In addition:

- Each target has its own target settings.
- There is only one current target at all times.
- The name of the current target appears in the Project window's tool bar.

Targets View Columns

The Targets view contains two columns of information. They are:

Targets–identifies the various targets available within the project.

Link Against Output–identifies the targets to precompile for the currently chosen target.

Creating Targets

You can create additional targets in a project.

To Create a Target

1. Choose **Create New Target** in the **Project** menu.

   The Create Target dialog box appears.
2. Enter `FirstTarget` in the **Create Target** dialog.
3. Choose either the **New Empty Target** setting, **OR**
   Clone Existing Target setting.

   If Clone Existing Target was chosen, specify which target setting to clone.
4. Click **OK**.

   The target `FirstTarget` appears in the Target list.

Renaming Targets

You can rename a target to better identify it. The target name is part of the target's individual settings, so you use the Target Settings dialog box to change the name.

To Rename a Target

1. Double-click the `FirstTarget` target to rename, **OR**
   Select the `FirstTarget` target and click the **Target Settings** icon**, OR**
   Select the `FirstTarget` target and choose **FirstTarget Settings** from the **Edit** menu.
2. Choose the **Target Settings** panel in the **Target Settings** list.
3. Enter a new name in the **Target Name** field. Try "AnotherTarget."
4. Click **Save**.

   The selected target is renamed and appears in the Target view.

Deleting Targets

During the course of programming, a target may become obsolete. When that occurs it's possible to delete it from the project quickly and easily.

To Delete a Target:

1. Select the `AnotherTarget` target to delete.
2. Choose **Remove Selected Items** from the **Project** menu, **OR**
   Type the keyboard shortcut Ctrl/Option+Delete.

   The `AnotherTarget` target disappears from the Target list.

NOTE: This action is not undoable.

Summary

This concludes the Targets view stage of this tour.

In this stage you learned how to:

- Identify the Targets view
- Identify the Targets view columns
- Create, rename, and delete targets

In the next stage you explore the Project window's toolbar.

Touring Toolbars

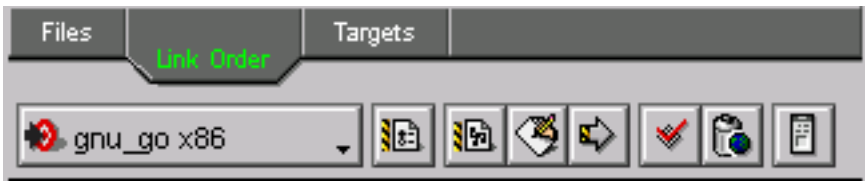
A toolbar provides buttons for common tasks. Each button contains an icon representing its action, and this action typically equates to a command found on the CodeWarrior menus.

The toolbar may appear in two places:

- Window Toolbar
- Floating Toolbar

Window Toolbar

Several windows within the CodeWarrior environment contain toolbars, including the Project window. The default Project window toolbar includes seven buttons and the Targets pop-up.



Showing and Hiding the Window Toolbar

To show or hide the Window toolbar in the Project window:

To Show the Window Toolbar:

- Choose **Window->Toolbar->Show Window Toolbar**

To Hide the Window Toolbar:

- Choose **Window->Toolbar->Hide Window Toolbar**

NOTE: The Project window must be active.

Default Button Actions

The default buttons that appear in the Window toolbar (from left to right) include:

Target Pop-up-sets and displays the current target.

Target Settings–displays the Target Settings window.

Bring Up To Date–compiles all the changed files in the project without creating any binary output.

Make–compiles and links all the files in the project and produces a binary file.

Run–compiles and links all the files in the project, produces a binary file, and then launches the binary file if it is an application.

Synchronize Modification Dates–updates the modifications dates stored in the project file, marking files with newer dates as requiring compilation.

Remove Object Code–removes all binary code from the selected targets.

Property Inspector–displays the Property Inspector window.

Floating Toolbar

The Floating Toolbar appears in a palette window that can be positioned anywhere on the screen for convenient use. In its default state, the Floating toolbar contains 22 buttons for accessing common commands.

To Show the Floating Toolbar:

- Choose **Window->Toolbar->Show Floating Toolbar**

To Hide the Floating Toolbar:

- Choose **Window->Toolbar->Hide Floating Toolbar**

Windows

Mac OS

Summary

This concludes the Touring Toolbars stage of this tour.

In this stage you learned how to:

- Identify the toolbars
- Identify the default buttons on the toolbars
- Hide and show the toolbars

NOTE: The toolbars are customizable, see the IDE User Guide for information.

Project Manager Actions

This section of the tutorial walks you through many common Project Manager actions.

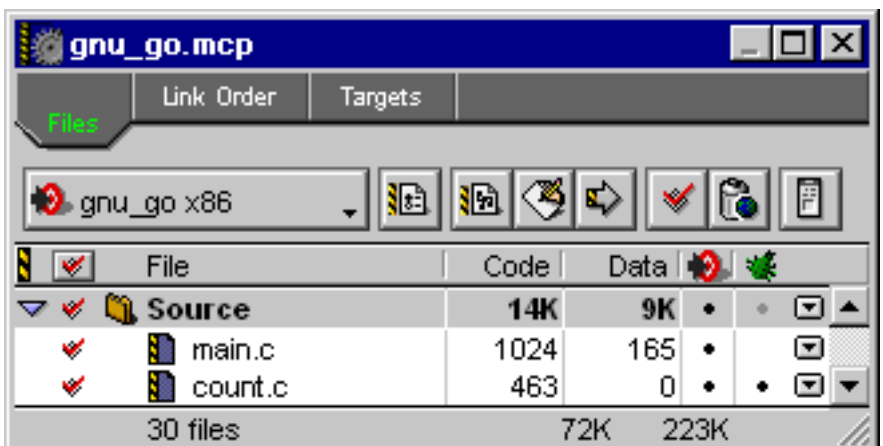
The topics covered include:

- Stage 5: Using File Actions
- Stage 6: Using Project Actions

Getting Ready

You will use the `gnu_go.mcp` project file throughout this tutorial. If you don't already have it open, please open the project now. The `gnu_go.mcp` project file is located in the `Source Examples` folder inside the `Core Tutorials` folder.

If you don't know how to open a project file, see the Guided Tour tutorial.



Using File Actions

In this stage you learn how to manage the files in a project.

The file actions include:

- Adding Files Manually
- Adding Files from an Editor Window
- Adding Files with Drag-and-Drop
- Renaming Files
- Removing Files
- Inspecting File Information
- Touching Files
- Compiling Files

Adding Files Manually

You must add a file to a project before the project can compile and link it. In this step you add the `foo_1.c` file.

To Add a File:

1. Choose **Add Files** from the **Project** menu.
2. Select the `foo_1.c` file in the open file dialog box.

   `foo_1.c` is located in the `ProjMgr` folder in the `Source Examples` folder.
3. Click **Add**.
4. (Mac OS only) Repeat Steps 2 and 3 for each file to add to the project.

   In this case, you have no more files to add.
5. (Mac OS only) Click **Done**.

   The file is added to the Project. If an "adding new access path" alert appears, you can safely ignore it.

Adding Files from an Editor Window

You must add a file to a project before the project can compile and link it. In this step you add the `foo_2.c` file.

To Add a File Window:

1. Choose **Open** from the **File** menu.

   A standard open dialog box appears.
2. Locate and select the `foo_2.c` file.

   `foo_2.c` is located in the `ProjMgr` folder in the `Source Examples` folder.
3. Click **Open**.

   An Editor window opens containing the file.
4. Choose **Add Window** from the **Project** menu.

   The `foo_2.c` file in the frontmost Editor window is added to the Project.

Adding Files with Drag-and-Drop

You can also drag a file into a project. In this step you add the `foo_3.c` file.

To Add a File(s) by Dragging:

1. Select the `foo_3.c` file to drag to the project.

   `foo_3.c` is located in the `ProjMgr` folder in the `Source Examples` folder.

2. Drag the `foo_3.c` file to the Project window.

   The `foo_3.c` file is added to the project at the position indicated by the focus bar.

Renaming Files

You can rename source files stored in a project.

To Rename a File:

1. Double-click the file to rename, **OR**
   Select the file and press the Enter/Return key.
2. Choose **Save As** from the **File** menu.
3. Enter a new name for the file in the save dialog box.
4. Click **Save**.

   The selected file is renamed in the Project window and a new copy of the file with the new name is saved to disk.

Removing Files

As files become obsolete, it becomes necessary to remove them from the project. At some point you may wish to remove a file from the project. In this step you remove the `foo_3.c` file.

To Remove a File(s):

1. Select the `foo_3.c` file.
2. Choose **Remove Selected Items** from the **Project** menu, **OR**
   Click the **Remove Selected Items** button on a toolbar, **OR**
   Type the keyboard shortcut Alt/Option+Delete.

   The selected file(s) disappear from the Project window.

NOTE: If a file is removed from the File view, it is removed from all targets in the project. If the file is removed from the Link Order view, it is removed only from the current target.

Inspecting File Information

There is additional information associated with every file in a project that is not displayed in the project window. You can view information about a file's attributes and targets in the Project Inspector window In this step you examine information for the `main.c` file.

To Inspect File Information:

1. Select the `main.c` file.
2. Choose **Project Inspector** from the **Window** menu, **OR**
   Click the **Project Inspector** button on a toolbar.

   The Project Inspector window appears with information about the `main.c` file.

Touching Files

To force a file to be compiled the next time you build a project, you 'touch' the file. You can also 'untouch' a file to remove it from the compiler list. In this step you touch the `main.c` file.

To Touch a File:

- Click inside the Touch column to the left of the `main.c` file, **OR** Choose **Touch** from the `main.c`'s interface pop-up menu.

  A marker appears in the Touch column.

To Untouch a File:

- Click inside the Touch column to the left of the `main.c` file, **OR** Choose **Untouch** from the `main.c`'s interface pop-up menu.

  The marker in the Touch column disappears.

Compiling Files

Use **Compile** to compile a selected file in a project. This is handy for those situations where you have modified one file and don't want to recompile the entire project. In this step you compile the `main.c` file.

To Compile a File:

1. Select the `main.c` file.
2. Choose **Compile** from the **Project** menu, **OR**
   Type the keyboard shortcut Ctrl+F7/Command+K.

   The `main.c` file is compiled.

Summary

This concludes the Using File Actions portion of this tour.

In this stage you learned how to:

- Add files manually, from an Editor window, and by dragging
- Rename and remove files from a project
- Get additional file information
- Touch and untouch files for compiling
- Compile individual files

In the next stage you explore Project actions. The are actions that apply to projects as a whole, not just to individual files.

Using Project Actions

In this stage you learn how to use the Project Manager to create, build, and close projects.

The project actions include:

- Creating a New Project
- Checking Syntax
- Building a Target
- Removing Project Object Code
- Closing a Project

Creating a New Project

To begin writing software, you need to create a project file to store all the sources, library, and other files associated with the software.

To Create a Project:

1. Choose **New Project** from the **File** menu.
2. Choose a Stationery type.

   If you are using Windows choose: `Win32-x86\C, C++\C Console App`.

   On Mac OS choose: `MacOS:C/C++:ANSI C Console 68k`.

   Leave the **Create Folder** check box enabled.
3. Click **OK**.
4. Enter a project name in the save dialog box. Use "`mw_go.mcp`."
5. Click **Save**.

   The new project folder and project are created and stored on disk.

Bringing mw_go.mcp Up to Date

A new project is useless without source files. To continue, you must remove one file, then add other source files to the `mw_go.mcp` project.

1. Remove the default source file from the `Source` group.

   On Windows, remove the `main.c` file from the Source group.

   On the Mac OS, remove the `HelloWorld.c` file.

   See Removing Files if you are unsure of how to remove a file.

2. Add the gnu_go source files to the `mw_go.mcp` project.

   The source files are located in the `gnu_go` folder inside the `Source Examples` folder. Add all the files with the `.c` suffix. Ignore the `.h` and plain text files. Place them in the `Sources` group of the project.

   See Adding Files Manually, Adding Files from an Editor Window, and Adding Files with Drag-and-Drop for help in adding files.

more…

Bringing mw_go.mcp Up to Date (continued)

3. Rename the target to better identify it.

   On Windows, use "`mw_go x86`."

   On the Mac OS, use "`mw_go_68k`."

   See Renaming Targets if you need assistance renaming a target.

With these steps completed, continue on.

Checking Syntax

Use the **Check Syntax** command to check for syntax errors in the selected file or group in the Project window. No object code is generated. In this step you will check the syntax of all the source files in the `Sources` group.

To Check Syntax:

1. Select the `Sources` group in the Project window.
2. Choose **Check Syntax** from the **Project** menu, **OR**
   Type the keyboard shortcut Ctrl/Command+; (semi-colon).

   The files in the `Source` group are checked for syntax errors. If any errors are found, the Errors & Warnings window appears listing them.

NOTE: By default, new C/C++ projects have the Require Function Prototypes option enabled. Ignore any prototype errors for now, we'll see how to deal with them later.

Building a Target

A project contains one or more targets. You build a target in order to create your software. That target may rely on other targets in the same project, but the Project Manager handles the interdependencies automatically.

If the target generates an application, you can compile and link the target using the **Run** command. In this step, you will build the new target.

To Build a Target:

- Choose **Run** from the **Project** menu, **OR**
  Click the **Run** button in a toolbar, **OR**
  Type the keyboard shortcut Shift+F5/Command+R.

  The project is compiled, linked, the object code saved to disk, and the application launched.

Removing Project Object Code

You can remove all the object code associated with a target in a project. This will force a complete rebuild or the project. In this step, you will remove the binary code from the target.

To Remove a Target's Object Code:

1. Choose the current target.

   On Windows, choose `mw_go x86`. On the Mac OS, choose `mw_go_68k`.

2. Choose **Remove Object Code** from the **Project** menu, **OR**
   Click the **Remove Object Code** button in a toolbar, **OR**
   Type the keyboard shortcut Ctrl/Command+– (hyphen).

3. Click **All Targets** to remove all the object code, **OR**
   Click **Current Target** to remove only the current target's object code.

   The project's Code and Data column totals are reset to zero because the compiled binary code stored in the project is removed.

Closing a Project

When you are finished working with a project, use one of the following techniques to close the project.

To Close a Project:

- Choose **Close** from the **File** menu, **OR**
  Click the Project window's close box, **OR**
  Type the keyboard shortcut Ctrl/Command+W.

  The Project window closes.

Summary

This concludes the Project Actions stage of this tutorial.

In this stage you learned how to:

- Create a project
- Touch files for compiling
- Check the syntax for multiple files
- Build a target
- Remove object code from a project
- Close projects

Conclusion

Congratulations!

You have just completed the CodeWarrior Project Manager tutorial. You have met the interface elements of the Project Manager, and learned how to use them to create projects, add files, and build applications. The Project Manager is another step on the road towards becoming a successful code warrior.

Now that you understand the Project Manager, its time to move on to another powerful component in CodeWarrior, the IDE Editor.

Technical Support

If you should encounter any difficulties with this guided tour, contact Metrowerks Technical Support for assistance. They can be reached at:

`support@metrowerks.com`

Copyright Information

Metrowerks CodeWarrior copyright ©1993–1998 by Metrowerks Inc. and its licensors. All rights reserved.

Documentation stored on the compact disk(s) may be printed by licensee for personal use. Except for the foregoing, no part of this documentation may be reproduced or transmitted in any form by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from Metrowerks Inc.

Metrowerks, the Metrowerks logo, CodeWarrior, and Software at Work are registered trademarks of Metrowerks Inc. PowerPlant and PowerPlant Constructor are trademarks of Metrowerks Inc.

All other trademarks and registered trademarks are the property of their respective owners.

ALL SOFTWARE AND DOCUMENTATION ON THE COMPACT DISK(S) ARE SUBJECT TO THE LICENSE AGREEMENT ON THE CD.

Revision Code: 1/14/98-lft